UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/024,961	12/19/2001	David Pociu	13729-003001	3454

26161   7590   08/20/2004

FISH & RICHARDSON PC
225 FRANKLIN ST
BOSTON, MA 02110

EXAMINER
MOFIZ, APU M

ART UNIT	PAPER NUMBER
2175	

DATE MAILED: 08/20/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

	Application No.	**Applicant(s)**
Office Action Summary	10/024,961	POCIU, DAVID
	Examiner	**Art Unit**
	Apu M Mofiz	2175

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE <u>3</u> MONTH(S) FROM
THE MAILING DATE OF THIS COMMUNICATION.
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed
  after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
  Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any
  earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *19 December 2001*.

2a)☐ This action is **FINAL**.   2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is
     closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-34* is/are pending in the application.

     4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-27* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☒ Claim(s) *28-34* are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *19 December 2001* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

     Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

     Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

     a)☐ All   b)☐ Some * c)☐ None of:

        1.☐ Certified copies of the priority documents have been received.

        2.☐ Certified copies of the priority documents have been received in Application No. _____.

        3.☐ Copies of the certified copies of the priority documents have been received in this National Stage
             application from the International Bureau (PCT Rule 17.2(a)).

     * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1)☒ Notice of References Cited (PTO-892)

2)☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3)☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
    Paper No(s)/Mail Date *4*.

4)☐ Interview Summary (PTO-413)
    Paper No(s)/Mail Date. _____ .

5)☐ Notice of Informal Patent Application (PTO-152)

6)☐ Other: _____.

DETAILED ACTION

Election/Restrictions

1.   Restriction to one of the following inventions is required under 35 U.S.C.

121:

    I.   Claims 1-27 are drawn to a distributed access system, where data

        requests are sent to a server in an xml format and the response is

        sent back in xml format to the client, classified in class 707,

        subclass 10.

    II.   Claims 28-34 are drawn to a GUI system or a visual program,

        where a plurality of visual controls is displayed, classified in class

        345, subclass 764.

2.   The inventions are distinct, each from the other because of the following

reasons:

Inventions in Groups I and II are distinct from each other if they are shown

to be separately usable. In the instant case, invention in Group I has separate

utility such as where data requests are sent to a server in an xml format and the

response is sent back in xml format to the client. See MPEP § 806.05(d).

Invention in Group II has separate utility and requires at least a visual program,

where a plurality of visual controls is displayed.

Inventions in Group I and II are unrelated. Inventions are unrelated if it

can be shown that they are not disclosed as capable of use together and they

have different modes of operation, different functions, or different effects (MPEP

§ 806.04, MPEP § 808.01). In the instant case the different inventions are a

distributed access system, where data requests are sent to a server in an xml

format and the response is sent back in xml format to the client (2) a GUI system

or a visual program, where a plurality of visual controls is displayed.

Because these inventions are distinct for the reasons given above and

have acquired a separate status in the art as shown by their different

classification, restriction for examination purposes as indicated is proper.

During a telephone conversion with Kenneth F. Kozic, Reg. No. 36,572 on

9th August 2004 a provisional election was made without traverse to prosecute

the invention of Group I, claims 1-27. Applicant in replying to this office action

must make affirmation of this election. Claims 28-34 are withdrawn from further

consideration by the examiner, 37 CFR 1.142(b), as being drawn to a non-

elected invention.

Applicant is reminded that upon the cancellation of claims to a non-elected

invention, the inventorship must be amended in compliance with 37 CFR 1.48(b)

if one or more of the currently named inventors is no longer an inventor of at

least one claim remaining in the application. Any amendment of inventorship

must be accompanied by a request under 37 CFR 1.48(b) and by the fee

required under 37 CFR 1.17(i).

Claim Rejections - 35 USC § 102

3.   The following is a quotation of the appropriate paragraphs of 35

U.S.C. 102 that form the basis for the rejections under this section made in this

Office action:

A person shall be entitled to a patent unless –

(a) the invention was known or used by others in this country, or patented or described in a printed publication in this or a foreign country, before the invention thereof by the applicant for a patent.

4.   Claims 1-27 are rejected under 35 U.S.C. 102(a) as being anticipated by

BEA (WebLogic Server 5.1 documentation, copyright 1997-2000 and last

updated 2/9/2000 and BEA hereinafter).

Examiner interprets the Applicant claimed invention as two system

exchanging data in XML format. XML is used because it is an industry standard.

One of the two systems (e.g. a client, one application, a Java client etc.) is

requesting some data (e.g. in case of a Java client, values for the attributes of a

Java object. It can be any data and the nature of the data depends the nature of

the applications and should be irrelevant) from another system. The second

system (e.g. a server, servlet, another application etc.) parses that request,

because it is in XML format to understand what the other system is requesting for

and then sending that request to another application (e.g. a EJB application or

any other application), which retrieves the requested data (e.g. from a database)

and transforms the results in XML format and sends it back to the requesting

application/system/client.

The BEA prior art clearly teaches all of the above features.

As to claim 1, BEA teaches a method comprising, in a network,

encapsulating data requests (i.e., StockClient is requesting for some transaction

by TraderBean and transforms the transaction request data in XML structure

before transferring/sending to StockServlet) (page 5; page 21) generated by an

application (page 5; page 21) in a first system (page 5; page 21); transferring the

encapsulated data requests to a second system (page 5; page 21); executing

(i.e. StockServlet operates/executes on TraderBean to execute the trade) (page

5; page 21) the encapsulated data requests (page 5; page 21) in the second

system (page 5; page 21); and processing (i.e. the StockServlet sends the

results back to StockClient in XML format for StockClient to process the data or

whatever it needs to do with that data) (page 5; page 21) in the first system (page

5; page 21) responses generated by the encapsulated (page 5; page 21) data

requests in the second system (page 5; page 21).

As to claim 2, BEA teaches that encapsulating comprises: generating an

Extensible Markup Language (XML) structure for each data request (i.e.,

StockClient is requesting for some transaction by TraderBean and transforms the

transaction request data in XML structure before transferring/sending to

StockServlet) (page 5; page 21); and converting the XML structure to an XML

request (page 5; page 21).

As to claim 3, BEA teaches the XML structure comprises a variable

stream (i.e. the input stream or output stream through which an application

receives the XML data; an application transferring data writes to an output

stream and an application receiving data reads data from an input stream.) (page

4) of data stored in memory of the first system (page 4), the stream including an

XML element (i.e. the data is delimited by a tag and called an element) (page 2)

for each request (page 5; page 21).

As to claim 4, BEA teaches the XML element (i.e. the data is delimited by a tag and called an element) (page 2) is a class object (i.e. instantiated objects of Java classes in Java client or Java server applications e.g. StockClient) (page 5; page 21) whose data is stored to generate XML (page 5; page 21).

As to claim 5, BEA teaches that the XML element (i.e. the data is delimited by a tag and called an element) (page 2) includes data from a data set object (i.e. instantiated objects of Java classes in Java client or Java server applications e.g. StockClient) (page 5; page 21).

As to claim 7, BEA teaches that transferring includes a text transmission protocol (i.e. HTTP) (page 4).

As to claim 8, BEA teaches that the text transmission protocol is Hypertext Transfer Protocol (page 4).

As to claim 9, BEA teaches de-encapsulating the encapsulated data requests by parsing into request statements (page 5; page 7; page 8; page 21); and executing the request statements (i.e. StockServlet operates/executes on TraderBean to execute the trade) (page 5; page 21).

As to claim 10, BEA teaches translating responses from the executed request statements into an XML format (page 5; page 21); and sending the XML formatted responses to the first system (i.e. the StockServlet sends the requested result data back to StockClient in XML format for StockClient to process the data or whatever it needs to do with that data) (page 5; page 21).

As to claim 11, BEA teaches a distributed application (page 5; page 21) method comprising: converting application requests in a first system (page 5; page 21); transmitting (page 5; page 21) the converted application requests to a second system over a network (page 5; page 21); parsing the converted application requests in the second system into request statements (page 5; page 7; page 8; page 21); and executing the request statements in the second system (page 5; page 7; page 8; page 21).

As to claim 12, BEA teaches that converting comprises: generating a data structure for storing data and parameters related to an application that produced the application requests (i.e. the request from the application e.g. attribute values of a Java object in a Java client, which it wants to retrieve from an application server) (page 5; page 21); translating (i.e., StockClient is requesting for some transaction by TraderBean and transforms the transaction request data in XML structure before transferring/sending to StockServlet) (page 5; page 21) the application requests into a standardized delimited data structure (i.e. the data is delimited by a tag and called an element) (page 2) stored in a memory of the first

system; and transforming the standardized delimited data structure in conjunction

with the data structure into a stream (i.e. the input stream or output stream

through which an application receives the XML data; an application transferring

data writes to an output stream and an application receiving data reads data from

an input stream.) (page 4) of text based data utilizing an Extensible Markup

Language (XML) format (page 5; page 21).

As to claim 13, BEA teaches that the parsing comprises: breaking down

the converted application requests to an executable command format utilizing

data and parameters related to an application (i.e. StockServlet

operates/executes on TraderBean to execute the trade) (page 5; page 21).

As to claim 14, BEA teaches that executing further comprises evaluating

executable commands prior to execution in the second system (i.e. StockServlet

operates/executes on TraderBean to execute the trade; The StockServlet or the

TraderBean has to look at/evaluate the sent parameters to know what it is

supposed to do.) (page 5; page 21).

As to claim 15, BEA teaches that executing further comprises evaluating

results generated by the executable commands (i.e. StockServlet

operates/executes on TraderBean to execute the trade; The StockServlet or the

TraderBean has to look at the sent parameters to know what it is supposed to do

with those parameters. After it executes whatever it needs to do, it has to look at/ evaluate the data that it is about to send back to the client) (page 5; page 21).

As to claim 16, BEA teaches converting the results into a stream of text-based data in a standardized XML format (i.e. the input stream or output stream through which an application receives the XML data; an application transferring data writes to an output stream and an application receiving data reads data from an input stream.) (page 4); and transmitting the converted results over the network to the first system (page 5; page 21).

As to claim 17, BEA teaches an application server (i.e. the EJBean server) (page 12) method comprising: generating a first data structure for storing data and parameters related to an application residing in the server (page 5; page 21); translating (page 5; page 21) application requests from the application into a delimited second data structure stored in a memory (i.e. the data is delimited by a tag and called an element) (page 2; page 5; page 21); generating a stream of text-based data in an Extensible Markup Language (XML) format from the second data structure (page 2; page 5; page 21).

As to claim 18, BEA teaches that the first data structure includes database tables (page 15; page 18), procedure results from logic calls and status/error messages (i.e. the TraderBean trading results are sent back to the StockClient via the StockServlet) (page 4; page 5; page 7; page 8; page 21).

As to claim 19, BEA teaches that the second data structure includes an element for each of the application requests (page 2; page 5; page 21).

As to claim 20, BEA teaches that the element (i.e. the data is delimited by a tag and called an element) (page 2) is a class object (i.e. instantiated objects of Java classes in Java client or Java server applications e.g. StockClient) (page 5; page 15; page 21) whose data is stored to generate XML (page 5; page 21).

As to claim 21, BEA teaches in a server, receiving a stream of text-based data in an Extensible Markup Language (XML) format (page 4; page 5; page 21); parsing the stream into request statements (page 4; page 5; page 7; page 8; page 21); and executing each of the request statements (page 4; page 5; page 21).

As to claim 22, BEA teaches that executing further comprises intercepting the request statements prior to execution and applying additional logic (i.e. the TraderBean receives the data from StockServlet and then applies trading related logic to those data) (page 4; page 5; page 7; page 8; page 21) based on a type or content of the request statements (page 4; page 5; page 7; page 8; page 21).

As to claim 23, BEA teaches that executing further comprises applying additional logic to responses generated from executing the request statements (page 4; page 5; page 7; page 8; page 21).

As to claim 24, BEA teaches converting responses generated from each of the executed request statements into an XML format (page 4; page 5; page 7; page 8; page 21).

As to claim 25, BEA teaches a computer program product residing on a computer readable medium having instructions stored thereon which, when executed by the processor, cause the processor to: convert application requests in a first system (page 4; page 5; page 7; page 8; page 21); transmit the converted application requests to a second system over a network (page 4; page 5; page 7; page 8; page 21); parse the converted application requests in the second system into request statements (page 4; page 5; page 7; page 8; page 21); and execute the request statements in the second system (page 4; page 5; page 7; page 8; page 21).

As to claim 26, BEA teaches a computer program product residing on a computer readable medium having instructions stored thereon which, when executed by the processor, cause the processor to: generate a first data structure for storing data and parameters related to an application residing in the server (page 4; page 5; page 7; page 8; page 21); translate application requests

from the application into a delimited second data structure stored in a memory

(page 2; page 4; page 5; page 7; page 8; page 21); generate a stream of text-

based data in an Extensible Markup Language (XML) format from the second

data structure (page 4; page 5; page 7; page 8; page 21).

Claim Rejections - 35 USC § 103

5.   The following is a quotation of 35 U.S.C. 103(a) which forms the basis for

all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described
> as set forth in section 102 of this title, if the differences between the subject matter sought to
> be patented and the prior art are such that the subject matter as a whole would have been
> obvious at the time the invention was made to a person having ordinary skill in the art to which
> said subject matter pertains. Patentability shall not be negatived by the manner in which the
> invention was made.

6.   Claim 6 is rejected under 35 U.S.C. 103(a) as being unpatentable over

(WebLogic Server 5.1 documentation, copyright 1997-2000 and last updated

2/9/2000 and BEA hereinafter). The prior art of BEA has been discussed in the

above rejections.

As to claim 6, BEA teaches a method comprising, in a network,

encapsulating data requests (i.e., StockClient is requesting for some transaction

by TraderBean and transforms the transaction request data in XML structure

before transferring/sending to StockServlet) (page 5; page 21) generated by an

application (page 5; page 21) in a first system (page 5; page 21); transferring the

encapsulated data requests to a second system (page 5; page 21); executing

(i.e. StockServlet operates/executes on TraderBean to execute the trade) (page

5; page 21) the encapsulated data requests (page 5; page 21) in the second

system (page 5; page 21); and processing (i.e. the StockServlet sends the

results back to StockClient in XML format for StockClient to process the data or

whatever it needs to do with that data) (page 5; page 21) in the first system (page

5; page 21) responses (i.e. The TraderBean results, error messages) generated

by the encapsulated (page 5; page 21) data requests in the second system

(page 5; page 21). BEA also teaches that the first or second system can be an

Object Oriented Java application (page 5; page 21). BEA teaches that the BEA

system can use a DBMS (page 15; page 18).

BEA does not show that the data set object includes table dictionaries,

column names, data from record sets and stored procedure parameters. The

nature of the attributes of a client application object will depend on the nature of

the client business and also what the client wants from the application server.

Whatever names/parameters/values the client want to transfer to the application

server, the BEA system has the ability to transfer using a text protocol e.g. HTTP.

The content description of data (i.e. the attributes of an object) will be different for

different applications.

Therefore these differences are only found in the non-functional

descriptive material and do not alter the function of data encapsulation at the first

system and de-encapsulation at the second system. Thus this descriptive

material will not distinguish the claimed invention from the prior art in terms of

patentability, see in re Gulack, 703 F.2d 1381, 1385, 217 USPQ 401, 404 (Fed

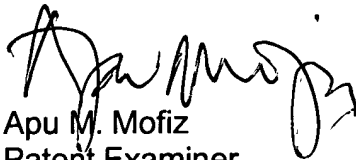Cir. 1983); In re Lowry, 32 F.3d 1579, 32 USPQ2d 1031 (Fed. Cir. 1994).

Therefore, it would have been obvious to a person of ordinary skill in the

art at the time of the invention was made to encapsulate the data at the first

system and de-encapsulate the data at the second system using the BEA

system. The subjective interpretation of the data does not patentably distinguish

the claimed invention.

Points of Contact

7.   Any inquiry concerning this communication or earlier communications from

the examiner should be directed to Apu M. Mofiz whose telephone number is

(703) 605-4240. The examiner can normally be reached on Monday – Thursday

8:00 A.M. to 4:30 P.M.

If attempts to reach the examiner by telephone are unsuccessful, the

examiner's supervisor, Dov Popovici can be reached at (703) 305-3830. The fax

numbers for the group is (703) 872-9306.

Any inquiry of a general nature or relating to the status of this application

should be directed to the Group receptionist whose telephone number is (703)

305-9600.

Apu M. Mofiz
Patent Examiner
Technology Center 2100

August 19, 2004